To Update or not: Dynamic Traffic Classification for High Priority Traffic in Wireless TSN

Jetmir Haxhibeqiri
IDLab, Ghent University – imec
9052 Ghent, Belgium
jetmir.haxhibeqiri@ugent.be

Xianjun Jiao
IDLab, Ghent University – imec
9052 Ghent, Belgium
xianjun.jiao@ugent.be

Pablo Avila Campos
IDLab, Ghent University – imec
9052 Ghent, Belgium
pabloesteban.avilacampos@ugent.be

Ingrid Moerman
IDLab, Ghent University – imec
9052 Ghent, Belgium
ingrid.moerman@ugent.be

Jeroen Hoebeke
IDLab, Ghent University – imec
9052 Ghent, Belgium
jeroen.hoebeke@ugent.be

***Abstract*—End-to-end low-latency deterministic communication, next to high-reliability communication, is one of the key features that communication systems are expected to provide for industrial systems. To achieve time-sensitive networking (TSN), a set of standards have already been designed and deployed for wired industrial communication systems, coexisting or replacing other long-living technologies such as Fieldbus, Profibus, or Modbus. Wireless time-sensitive networking (W-TSN) is getting traction with the development of the newest WiFi generation (IEEE 802.11be) as well as advances in cellular networking. One of the challenges in W-TSN is scheduling and isolation of time-critical traffic in the shared wireless medium. In this paper we present a solution, called dynamic traffic classification, to give faster dedicated access to the wireless medium for packets of highly-time-sensitive flows, that can be generated randomly. Dynamic traffic classification utilizes so-called shadow queues implemented in FPGA-based WiFi baseband SDR platform, openwifi, to prioritize channel access of certain packets over others. We show that the channel access latency in the case of dynamic traffic classification does not depend on the scheduling cycle, but on the distribution of dedicated time slots inside the schedule cycle. As such we achieve to decrease the end-to-end latency by 75% in case of longer communication cycles with wider space between communication time slots.**

***Index Terms*—Wireless time-sensitive networking, scheduling, dynamic classification, openwifi, WiFi**

I. Introduction

Time-Sensitive Networking (TSN) is increasingly important for industrial communication as it supports communication between industrial processes, machines, and people, facilitating fast production processes, reducing costs, and improving customization. TSN is a set of IEEE standards that enables deterministic communication over standard Ethernet networks. It standardizes a set of time-sensitive features over Ethernet: accurate time synchronization [1], traffic scheduling [2], frame preemption [3], stream policing [4], frame replication [5] and network management [6]. TSN provides the real-time communication and coordination required for applications in industrial control and automation, enabling time-sensitive, deterministic low latency communication. While wired TSN can fulfill industrial application requirements, it does not provide flexibility and mobility as made possible with wireless communication [7]. As such, despite the challenges associated with wireless, wireless TSN (W-TSN) is getting traction for industrial applications.

W-TSN can be achieved using 5G cellular networks where wired TSN networks are bridged using 5G-TSN logical bridge [8]. With the advancement of the feature set that can be used in the seventh WiFi generation, IEEE 802.11be, such as multi-link operation, coordinated OFDMA and coordinated spatial re-use (C-SR) [9], WiFi is becoming increasingly important for W-TSN application as well. The main two challenges of W-TSN are accurate time synchronization and traffic scheduling over shared medium. While accurate time synchronization can be achieved with accuracy down to microsecond level [10], traffic scheduling is only performed at run time with fewer possibilities of updating over the operational time.

A schedule update for a single time slot in one node in W-TSN could lead to the overall schedule update for each node connected to the same access point (AP). This comes due to the shared nature of the wireless medium, where extending schedule of one node might impact the transmissions of the other. The single node schedule update is only possible when there is still unused network capacity (sub-carriers, time slots, etc) that can be assigned to that node. However, still, the application requirement might not be fulfilled due to the organization of the schedule in the WiFi cell.

In order to maintain application requirements, one possibility is to update the traffic classification utilizing the currently available schedule. In this paper we consider static communication schedules in one WiFi cell, however, we do dynamic traffic classification between different queues to fulfill the application needs. We develop such feature in FPGA-based WiFi baseband software defined radio (SDR) platform, openwifi, and show that dynamic traffic classification maintains latency requirements of the time-critical traffic flows even under heavy load of other traffic in the network. We compare the proposed solution to static traffic classification and to the case with the absence of additional shadow queues in the physical layer. Contributions of this paper include:

- dynamic traffic marking of packets
- dynamic traffic classification method implemented in openwifi driver
- additional hardware queues to support faster channel access for re-classified packets
- evaluation in a real test-bed environment

This paper is organized as follows. In section II we discuss related work to scheduling in TSN and W-TSN. Section III gives a technical background to W-TSN and its realization in openwifi SDR platform as well as the problem statement related to scheduling in W-TSN. Section IV details the dynamic classification feature design and implementation. Section V shows the evaluation of the dynamic traffic classification with and without shadow queues in terms of achieved latencies of time-critical traffic while section VI concludes the paper.

II. Related Work

Two main mechanisms for traffic scheduling in wired TSN relate to time-aware shaping (IEEE802.1Qbv) [2] and credit-based shaping (IEEE802.1Qav) [11]. However, scheduling traffic in an end-to-end fashion entails computation of each schedule in each node and alignment between schedules in a multi-hop network to provide the requested latency requirement by the network. Such a problem is shown to be an NP-hard problem [12].

In [13] authors present the no-wait packet scheduling problem for time-aware shaping. The scheduling problem in [13] is formulated as an integer linear programming (ILP) problem, with an alternative for Tabu search based heuristic for increased scalability of the solution. Similarly, authors in [14] address routing and scheduling in multi-hop wired TSN as an iterated integer linear programming problem.

In [15] authors study the combinability problem of different time-sensitive flows in a multi-hop TSN network. They present a scheduling method that is based on non-collision theory of such flows and dynamic scheduling of best effort traffic flows. Differently to our work, the dynamic traffic scheduling in [15] is used for best effort traffic, whereas in our research we use it to fulfill latency requirements of the time-sensitive flows. Moreover, the presence of additional shadow queues in our solution makes the dynamic traffic classification and scheduling independent of the number of packets in the current queues. This approach will avoid head-of-line blocking for time sensitive traffic by other best effort traffic.

A specific type of time-sensitive flows are event-driven flows that are generated randomly at any time. In [16] authors present a traffic management scheme for wired TSN bridges and end nodes that gives explicit support for event-driven real-time traffic. In [17] authors present a joint algorithm that considers packet fragmentation as well as no-wait scheduling under hard real-time constraints. Differently from [17], in this work we consider only dynamic traffic classification on each network hop, to decrease the overall end-to-end latency.

In [18] authors present a deadline-driven scheduling approach for time-sensitive flows in wired TSN. They make use of Per-Stream Filtering and Policing (PSFP) mechanism from [19] to change the traffic flow priority based on the applied schedule. The traffic priority is increased as the deadline approaches. In wireless TSN, increasing the traffic flow priority does not necessarily decrease the channel access time, if dedicated time slots for lower priority traffic are scheduled in the communication cycle. As such the dynamic traffic classification based on the time slot organization in the communication cycle is a must, as we show in this paper.

In addition to scheduling problem in wired TSN, there are some initial works on scheduling problems in wireless TSN. Authors in [20] present the Scheduled Transmission Opportunity (S-TXOP) concept that decreases the control overhead for scheduling in uplink and donwlink in IEEE 802.11ax. Such technique consist of slotting the transmission opportunities gained by the AP. At the beginning of each TXOP, AP will send the S-TXOP control frame to organize the TXOP time in slots for UL/DL transmissions from/to certain clients. In [21] authors present a joint end-to-end scheduling optimization problem based on both wireless and wired TSN constraints, using constrained programming. The proposed framework is validated in a simulator environment. In [22] authors present a centralized model for network management and mapping of different traffic flows from one domain (wired TSN) to the other (5G-based wireless TSN).

III. Background and Problem Statement

This section provides a short background on wireless time-sensitive networking (W-TSN) and dives into the problem of updating the schedules in W-TSN.

A. Wireless Time-Sensitive Networking

Time-Sensitive Networking (TSN) is a set of different standards that enable deterministic communication over Ethernet networks. These standards define needed network functionalities and features with the basic set being: end-to-end time synchronization, traffic scheduling, traffic classification, and TSN network management.

To have control over packet transmissions the first required network feature is network-wide accurate time synchronization. This feature provides the ability to all nodes in the network to have the same notion of time. This is achieved using the Precision Time Protocol (PTP) [1] , originally for wired networks, but extended to wireless as well [10]. PTP is based on two-way exchanges of synchronization packets between the time master and its slaves. The time master firsts announces its presence by transmitting *Sync* packets, while the impact of link transmission delays on time synchronization accuracy is resolved by transmissions of *DelayReq* and *DelayResp* packets between slave and master.

Next to time synchronization, nodes in the network should agree on the timing of transmissions of packets. Several scheduling approaches are used in TSN: such as time-aware shaper (TAS) [23], credit-based scheduler, cyclic queuing and forwarding, etc. TAS is used to separate time-critical traffic flow(s) from best-effort traffic flow(s) on a time basis. Time is

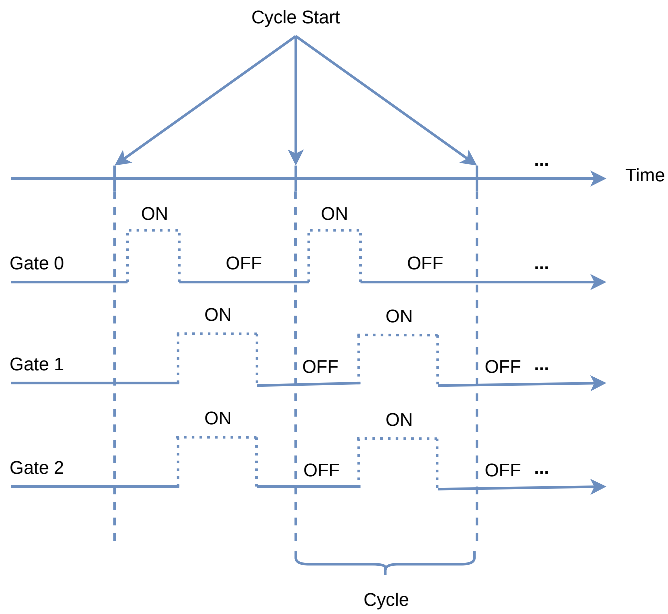


Fig. 1: Gating system for traffic scheduling in W-TSN

divided into communication cycles consisting of multiple gate openings of varying lengths. These openings determines which queue is selected to transmit packets. If multiple gates are opened simultaneously, packets are transmitted based on queue priority. To isolate time-critical traffic flow from other traffic flows, it's typically assigned to a queue with an exclusive gate opening during the cycle, not shared with other queues. This allows for better isolation of the time-critical traffic flow. Figure 1 illustrates the operation of a Gate Control List (GCL) in a TSN-enabled interface.

To achieve traffic scheduling, each traffic flow must be classified and directed to a specific queue. Several classifications are used in TSN. At the data link layer, packets are classified using VLAN tag IDs, while at the network layer IP packets are classified using Differentiated Services Code Point (DSCP) values. Always there is a limited number of queues compared to the possible values that either VLAN ID (12 bits) or DSCP (6 bits) can take. As such, in addition to the classification of packets, mapping functions between the classification identities to the hardware queues need to be enforced. Even if two different traffic flows can have different classification IDs they might end up in the same hardware queue, hence being treated similarly.

All of these three mechanisms have been implemented in openwifi [24], the first open-source WiFi software-defined platform. The time synchronization is achieved using PTP over wireless, where Time Synchronization Function (TSF) timer is utilized to support μs level synchronization [10]. In addition, openwifi provides 4 hardware queues, that can be controlled by a gated mechanism based on the same TSF timer. The communication cycle supported in openwifi is of the form $512 * 2^n$ μs for $n \in [0, 7]$, while time slots can be multiple of 128 μs long. Traffic classification is done based on the 2

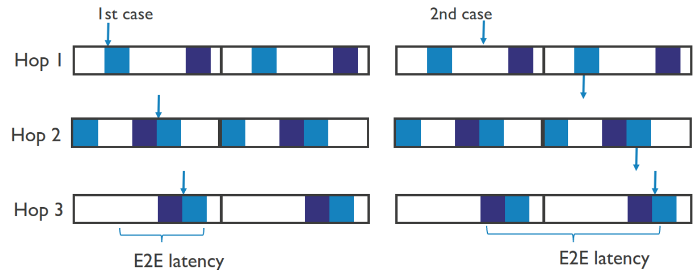


Fig. 2: End-to-end latency depends on the time packet is generated.

most significant bits of the DSCP value for IP traffic, and for layer 2 traffic only VLAN IDs are used.

B. Scheduling problem in W-TSN

In wireless communication, the channel is shared between nodes, in contrast to wired networks where each link is used only by its end link devices. When a new time-sensitive traffic flow is introduced in the network there is a need for a new dedicated time slot as well. This dedicated time slot can be taken from the time that is currently not used by any of the nodes in the cell. When this is not possible, a partial schedule update is required for the nodes that are affected by this traffic flow introduction. When there is no time window left to assign a new portion of time, nor possible partial updates of the schedule, the traffic flow must be scheduled in a shared time slot. However, this might not fulfill application requirements in terms of latency and throughput.

Next to schedule updates for new traffic flows, there is a challenge on how to fulfill traffic requirements, especially latency, for sparse and randomly generated traffic. Such traffic can be generated from random events such as a stop event for a robot-controlled machine in industry or a sensor reading of a critical event. One way to support such traffic is to reserve time slots, but this will cause an overall throughput decrease due to unused bandwidth.

Even when the bandwidth is reserved, the communication latency depends on the time from the moment when the traffic is generated until the reserved time slot will open. This is shown in Figure 2 for two cases. When the packet is generated before the time slot in the first hop, then the end-to-end latency is only the latency introduced by the time-slot organization in different hops. Assuming no-wait time organization of time-slots in different network hops, the end-to-end latency will consist only on the time difference between the start of the time slot in the first network hop and generation time, and summation of transmission times on each network hop. When the packet is generated after the time slot in the first hop, it has to wait until the next cycle to be transmitted. Thus the maximal communication latency is bounded based on the cycle length, in case of a single time slot per cycle, or maximal time distance between two consecutive time slots assigned to the same queue in the same cycle.

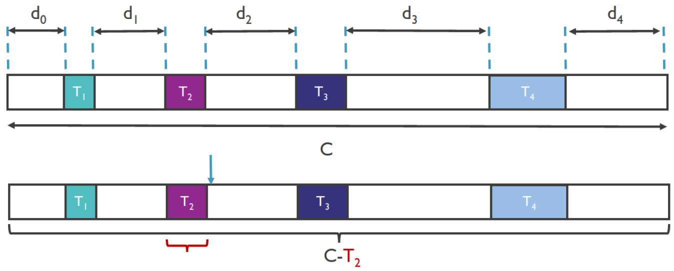


Fig. 3: Dynamic classification example model. Example of a schedule and end-to-end latency example when the packet is generated just after the time slot.

IV. Dynamic traffic classification design

In this section, we give a detailed description of how dynamic traffic classification can improve the latency requirements of the time-sensitive flows in the case they are randomly generated. We suggest that random low-latency traffic can be supported without the need for throughput reservations or by schedule updates on due time.

A. Approach

In current W-TSN approaches, traffic classification is done either based on VLAN ID tags or DSCP values. This means that a packet traversing the network will always be scheduled in the same queue and time slot in each hop, considering static mapping.

Dynamic traffic classification is based on the ability of the wireless interface driver to re-classify traffic flows on different queues and time slots on a per-packet basis no matter the actual tagging used. As such, the re-classification does not consider the DSCP or VLAN tag, but it considers only the relative delay from current moment until the next time slot assigned to the node. For example, if a packet is originally assigned to time slot T_3, but upon arrival at the driver, the earliest available time slot for that node is T_2, then the packet is re-classified and sent to the corresponding queue that is served at time slot T_2. As such, the dynamic traffic classification will be based on the position of the already assigned time slots inside the communication cycle, and not on static traffic classification mapping from DSCP values or VLAN tags to queues.

Consider the case when one wireless node has 4 queues with each of them having one time slot T_i, $1 \leq i \leq 4$, assigned in a communication cycle being C time units long as shown in Figure 3. Let the start of time slot T_i, $2 \leq i \leq 4$, be d_{i-1}, $2 \leq i \leq 4$, time units apart from the end of time slot T_{i-1}, $2 \leq i \leq 4$, and let d_0 be the distance of the start of time slot T_1 from start of communication cycle, and d_4 the distance of the end of time slot T_4 from end of communication cycle. Then communication cycle will be $C = \sum_{n=1}^{4} T_i + \sum_{n=0}^{4} d_i$. In order to achieve low end-to-end latency, the communication cycles in consecutive communication hops are aligned to support continuous transmissions without the need to wait on each hop. If dynamic classification is not used, the worst-case end-to-end latency occurs when the packet is generated just after the time slot of the queue to which it should be classified. In such a case, the packet has to wait for the full communication cycle for transmission to start at the scheduled time slot (known as residual time), as well as the transmission time on each hop. An example is shown in Figure 3. Thus, the worst-case end-to-end latency for a packet classified in the i-th queue that will be served at time slot i will be:

$$E2E_{latency} = (C - T_i) + h * ToA \tag{1}$$

where C is the communication cycle length, h is the number of network hops packet travels and ToA is the time on air of the packet which is the transmission time of the packet. Here we assume that processing time in the network stack of each node is negligible compared to transmission time on each hop. On the other hand, when dynamic traffic classification is used, each packet can be transmitted at each time slot. As such the worst-case end-to-end latency depends only on the maximum distance between the consecutive time slots. In this case for any dynamically classified packet end-to-end latency will be:

$$E2E_{latency} = max(d_i, d_0 + d_4) + h * ToA \quad 2 \leq i \leq 3 \tag{2}$$

where the $max()$ is the maximum operation of all inter time slot distances.

The end-to-end latency in Equation 2 assumes that the packet that is re-classified in the new queue is not blocked by head-of-line blocking in hardware queues of the interface. In that case, the packet transmission will be delayed until the transmission of the packets ahead in the queue. If the time slot is sufficiently long to start the transmission of the re-classified packets, then the impact on the end-to-end latency is negligible. If the packet is not transmitted in that communication cycle, then the end-to-end latency will increase by communication cycle length.

B. Implementation

Enabling dynamic classification at the driver level depends on the traffic flow sensitivity. If the traffic flow is highly sensitive to latency, then no matter when it is generated it has to be transmitted as soon as possible, overtaking the transmission opportunities of other traffic flows. As such the implementation is composed of three key features: marking of the traffic flows that need to be dynamically classified at user space, dynamic classification of the traffic flow at the driver level, and multiple-queue system at the hardware level to support it. All this implementation is done based on the openwifi SDR platform [25] and in-band network telemetry [26].

1) Dynamic classification marking of the data packets is done at user space based on the application requirements. Each application reports its own requirements and based on this information, the communication stack marks the filters of the traffic flow packets based on the source/destination IP/port and transport protocol used. Once packets are filtered from other traffic, they are marked with a dynamic classification

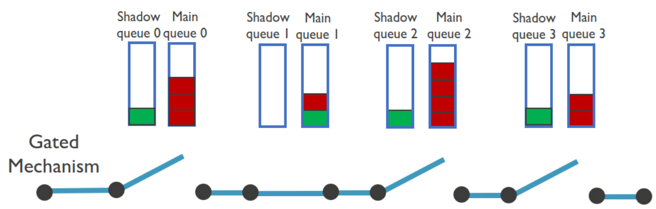


Fig. 4: Main and shadow hardware queues.

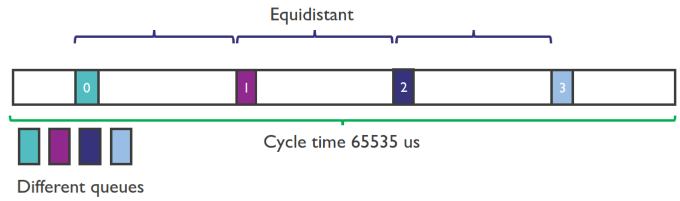


Fig. 5: Schedule used during experiments.

TABLE I: Test-bed setup parameters

Parameter	Value
Center frequency	5180 MHz
Bandwidth	20 MHz
Physical data rate	26 Mbps
Packet size	500 bytes
Time slot length	256 μs
Communication cycle length	65,535 ms
Experiment time	1 hour

flag that is added as an IPv6 extension header together with in-band network telemetry (INT) information as in [26]. INT in this case will be used only for performance measurements. However, dynamic classification flag is independent of INT, and can be present in the IPv6 packet even when INT is not enabled.

2) *Dynamic classification at driver level* is achieved by parsing the IPv6 extension header at a fixed offset to get the dynamic classification flag. If the packet has the dynamic traffic classification flag enabled, the current time offset inside the communication cycle is determined by performing a modulo operation of the current time with the communication cycle length. Then the next time slot available for that node is determined by doing a binary search in the sorted array of time slot start times. Once the next time slot is determined, the packet is sent to the determined hardware queue that is served during that time slot.

3) *openwifi hardware queues* implementation is based on four main hardware queues and four shadow queues as shown in Figure 4. Thus each main queue has one shadow queue that is served at the same time as the main queue on a priority basis. As such, a packet that is queued in the shadow queue will take priority in being transmitted compared to a packet that is queued in the main queue. This way, the head-of-line blocking of the dynamically classified packet by a packet already residing in the queue is avoided. Of course, the shadow queues will be used only for the dynamically classified packets and not for normal traffic flows. The classification mechanism in the driver must send the re-classified packets to a shadow queue only.

V. Results

In this section, we present the results when dynamic traffic classification is used without shadow queues, as well as when shadow queues are employed.

A. Evaluation setup and experiments

The test setup is composed of two W-TSN openwifi enabled end nodes, one W-TSN openwifi access point, one TSN Linux switch, and one network controller. The network controller manages the setup, distributes the communication schedules, initiates the PTP master, collects the in-band network telemetry data, and shows the monitored information in real-time on a dashboard.

The channel used was set at 5180 MHz, the data rate of all nodes was fixed at 26 Mbps (MCS 4 in 802.11n) and the packet length of the packet was fixed at 500 bytes. All the other parameters used in the setup during the measurements are shown in Table I. In order to see the impact on latency when the dynamic traffic classification is enabled, the communication cycle was set to 65,535 ms assigning equidistant time slots of 256 μs for each queue. The same schedule is applied to all wireless nodes, as is shown in Figure 5. Nodes are kept time synchronized using PTP, which is scheduled in queue 0 with all the other WiFi control and management traffic (beacons, probes etc). The other three queues (1-3) are used for data traffic in downlink.

We introduce three different traffic flows in the network in downlink direction. When no dynamic traffic classification is used, each of the flows will be scheduled in queues 1 to 3. Flows scheduled in queue 1 and queue 3, are UDP traffic flows with data rate of 80 kbps with destination wireless client 2. Traffic flow scheduled in queue 2 is a periodic traffic flow generating one 500 bytes long packet every 65 ms, resulting in $\sim$ 62 kbps data rate, and is destined to client 1. The low data rate of the traffic flows comes as a fact of the schedule used. Most of the cycle time is not used for transmission thus the supported application data rate is low. However, this does not invalidate the dynamic traffic classification mechanism results. During the experiments only traffic flow scheduled in queue 2 will be dynamically classified.

B. Dynamic classification without shadow queues

We perform experiments for 1h by sending three different traffic flows in downlink from the access point towards end nodes. The end-to-end delay is measured using in-band network telemetry [27]. The dynamic classification is enabled only for the traffic flow scheduled in queue 2, while the other two traffic flows are not dynamically classified. We compare the end-to-end latency when the dynamic classification is enabled and when the dynamic classification is disabled for the traffic flow.

When the traffic flow is not dynamically classified, the end-to-end latency is bounded by the communication cycle length as shown in Figure 6a of ∼65 ms. When we enable the dynamic traffic classification, packets of traffic flow 2 can be transmitted at any time slot that is assigned to the AP. As such, based on the applied schedule the end-to-end latency is bounded by the maximal distance of the assigned time slots as shown in equation 2. In this case, the end-to-end latency is decreased to lower than ∼16 ms, 1/4 of the communication cycle as the number of time slots in one cycle is 4, as shown in Figure 6b. The peaks in the end-to-end latency shown in Figure 6b are due to head-of-line blocking since in this case, we are not using any shadow queues. Still, there is a possibility that in the queue there are other packets before the dynamically classified one. Note that the data absence of the latency measurements in Figure 6a and 6b, respectively, are due to the restart of the traffic generators during the experiment time.

From Figure 6c and 6d, we see that in 99% of cases, the latency is smaller than 65 ms and ∼23 ms, respectively. By doing the dynamic classification of traffic flows we have achieved to decrease the latency only to ∼ 1/4 of the communication cycle.

C. Dynamic classification with shadow queues

Similar to the previous case we perform measurements for 1h long. All 4 queues in the access point are used to send three different traffic flows in downlink, one traffic flow for each queue (1 to 3), while queue 0 is reserved for the PTP traffic and control and management traffic. The dynamic traffic classification is enabled only for traffic flow 2, which in the case of dynamic classification will be assigned to the shadow queue of the main queue that will be served next.

Figure 7 shows achieved end-to-end latency when the dynamic traffic classification is enabled, in presence of shadow queues. As in the previous case, the benchmark in absence of the dynamic classification is shown in Figure 7a. Even when shadow queues are present, they will not impact the latency as they are not used for normal traffic classification. Thus, the end-to-end latency is bounded by the cycle length. On the other hand, when the shadow queues are used together with dynamic traffic classification the end-to-end latency is decreased to the maximal time distance between time slots assigned to that node. As is shown in Figure 7b, the end-to-end latency is lower than 16 ms. The CDF of this latency shows that in 99% of the cases, the end-to-end latency is smaller than 17 ms (Figure 7d). This is much better compared to the case when no shadow queues were used (refer to Figure 6d).

VI. Conclusion and Future Work

Traffic scheduling and maintaining the requested communication latency for applications is a crucial part of time-sensitive networking. With extending of time-sensitive networking concepts to wireless networks, this becomes even more challenging. Due to the shared nature of the wireless medium, an update of the schedule of one node might impact other nodes' schedule and might require further updates to other nodes as well. As such, frequent updates of the traffic schedule in wireless time-sensitive networks should be avoided.

In this work, we showed how the latency requirements for certain traffic flows can be maintained by enabling dynamic traffic classification without the need of frequent updates of the schedule. The implementation of dynamic traffic classification was done in near to the physical layer (at driver level) of open-source SDR platform, openwifi, while traffic marking was done in user space. Moreover, to avoid the head-of-line blocking in hardware queues, we have implemented the so-called shadow queues, which are served at the same time as their counterpart main queues on a priority basis. As such no-matter if there are packets in the main queue, the dynamically classified packets in shadow queue will access the medium first.

Experiments were performed in a real-world testbed that was composed of openwifi enabled end nodes and an openwifi enabled access point. We demonstrated that when dynamic traffic classification is enabled the end-to-end latency does not depend on the communication cycle length, but on the maximal time distance of consecutive time slots assigned to that node, no matter the queue assignment. In an extreme case when the time slots per each of the 4 main queues are assigned in an equidistance, the end-to-end latency decreases by a factor of 4. Also, we showed that when no shadow queues are used, the end-to-end latency requirement is still in the same range, however, there are peaks in latency due to head-of-line blocking of the main queues. In case of usage of dynamic classification and shadow queues, the end-to-end latency dropped 4 times for the schedule under-test.

Acknowledgment

This research was partially funded by the imec ICON project VELOCe - VErifiable, LOw-latency audio Communication (Agentschap Innoveren en Ondernemen project nr. HBC.2021.0657), the Flemish FWO SBO S003921N VERI-END.com (Verifiable and elastic end-to-end communication infrastructures for private professional environments) project and the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

References

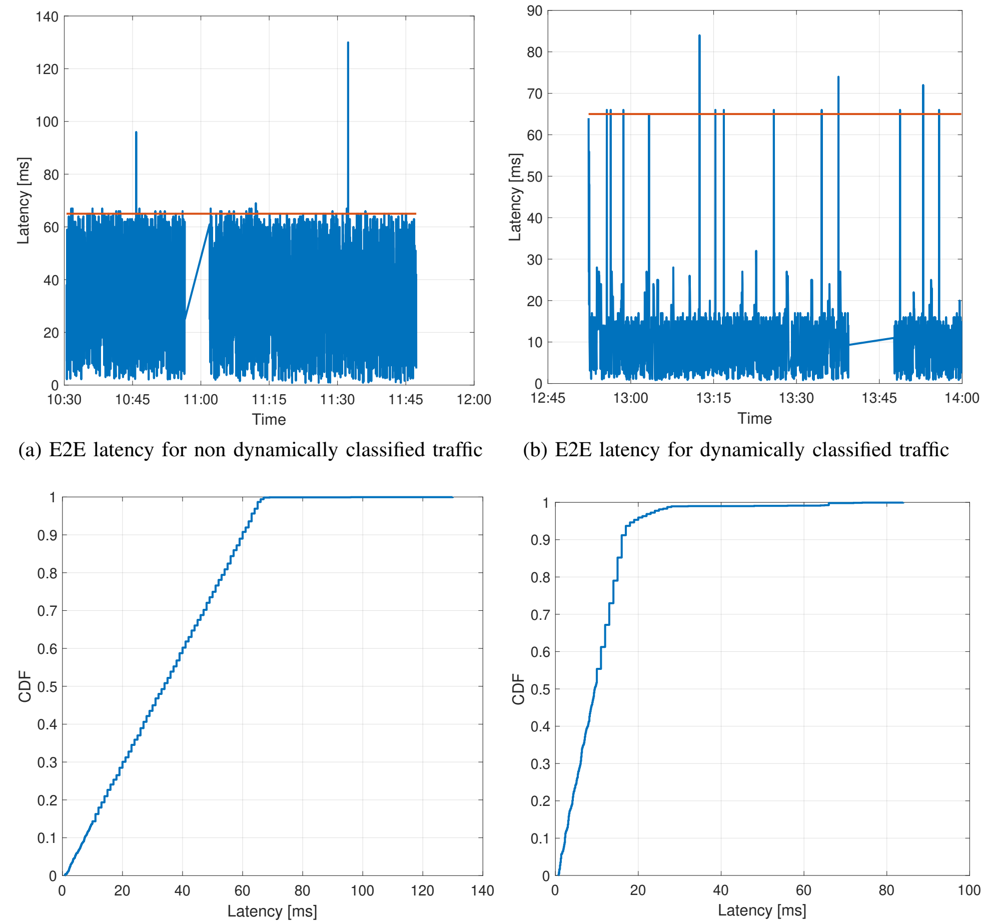
[1] *IEEE Standard 802.1AS-2020, "IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks"*, IEEE Standards Association, June 2020.

[2] "Ieee standard for local and metropolitan area networks – bridges and bridged networks - amendment 25: Enhancements for scheduled traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1–57, 2016.

[3] *IEEE 802.1Qbu-2016 "IEEE Standard for Local and metropolitan area networks, Bridges and Bridged Networks, Amendment 26: Frame Preemption"*, IEEE Standards Association, August 2016.

[4] *IEEE 802.1Qci-2017 "IEEE Standard for Local and metropolitan area networks, Bridges and Bridged Networks, Amendment 28: Per-Stream Filtering and Policing"*, IEEE Standards Association, September 2017.

[5] *IEEE 802.1CB-2017, "IEEE Standard for Local and metropolitan area networks-Frame Replication and Elimination for Reliability"*, IEEE Standards Association, October 2018.



(a) E2E latency for non dynamically classified traffic

(b) E2E latency for dynamically classified traffic

(c) CDF of E2E latency for non dynamically classified traffic

(d) CDF of E2E latency for dynamically classified traffic

Fig. 6: End-to-end latency and its cumulative distribution function (CDF) for non-dynamically classified traffic and dynamically classified traffic.

[6] *IEEE P802.1Qcc-2018, "Standard for Local and metropolitan area networks - Bridges and Bridged Networks - Amendment:Stream Reservation Protocol (SRP) Enhancements and Performance Improvements"*, IEEE Standards Association, October 2018.

[7] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation ethernet, iiot, and 5g," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[8] *Integration of 5G with Time-Sensitive Networking for Industrial Communications*, 5G-ACIA, White Paper, January 2021.

[9] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-sensitive networking in ieee 802.11 be: On the way to low-latency wifi 7," *Sensors*, vol. 21, no. 15, p. 4954, 2021.

[10] M. Aslam, W. Liu, X. Jiao, J. Haxhibeqiri, G. Miranda, J. Hoebeke, J. M. Marquez-Barja, and I. Moerman, "Hardware efficient clock synchronization across wi-fi and ethernet based network using ptp," *IEEE Transactions on Industrial Informatics*, 2021.

[11] "Ieee standard for local and metropolitan area networks– virtual bridged local area networks amendment 12: Forwarding and queuing enhancements for time-sensitive streams," *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pp. 1–72, 2010.

[12] S. S. Craciunas, R. S. Oliver, and T. Ag, "An overview of scheduling mechanisms for time-sensitive networks," *Proceedings of the Real-time summer school LÉcole dÉté Temps Réel (ETR)*, pp. 1551–3203, 2017.

[13] F. Dürr and N. G. Nayak, "No-wait packet scheduling for ieee time-sensitive networks (tsn)," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, 2016, pp. 203–212.

[14] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4525–4534, 2019.

[15] Y. Lu, L. Yang, S. X. Yang, Q. Hua, A. K. Sangaiah, T. Guo, and K. Yu, "An intelligent deterministic scheduling method for ultralow latency communication in edge enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1756–1767, 2022.

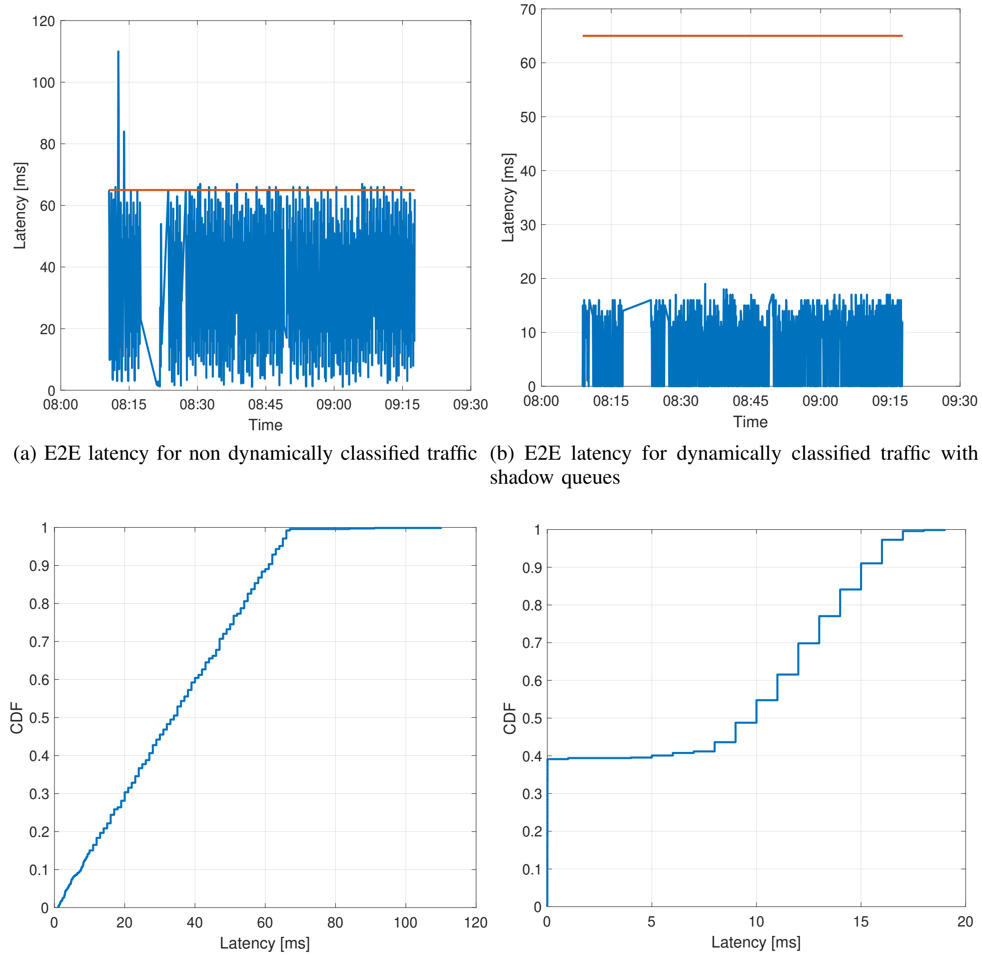
[16] L. Lo Bello, G. Patti, and G. Vasta, "Assessments of real-time communications over tsn automotive networks," *Electronics*, vol. 10, no. 5, p. 556, 2021.

[17] X. Jin, C. Xia, N. Guan, and P. Zeng, "Joint algorithm of message fragmentation and no-wait scheduling for time-sensitive networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 478–490, 2021.

[18] G. Patti, L. L. Bello, and L. Leonardi, "Deadline-aware online scheduling of tsn flows for automotive applications," *IEEE Transactions on Industrial Informatics*, 2022.

[19] "Ieee standard for local and metropolitan area network–bridges and bridged networks," *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1–1993, 2018.

[20] D. Akhmetov, D. Das, D. Cavalcanti, J. Ramirez-Perez, and L. Cariou, "Scheduled time-sensitive transmission opportunities over wi-fi," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 1807–1812.



(a) E2E latency for non dynamically classified traffic

(b) E2E latency for dynamically classified traffic with shadow queues

(c) CDF of E2E latency for non dynamically classified traffic

(d) CDF of E2E latency for dynamically classified traffic with shadow queues

Fig. 7: End-to-end latency and its cumulative distribution function (CDF) for non-dynamically classified traffic and dynamically classified traffic with shadow queues.

[21] D. Ginthör, R. Guillaume, J. von Hoyningen-Huene, M. Schüngel, and H. D. Schotten, "End-to-end optimized joint scheduling of converged wireless and wired time-sensitive networks," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 222–229.

[22] Z. Satka, I. Álvarez, M. Ashjaei, and S. Mubeen, "Work in progress: A centralized configuration model for tsn-5g networks," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2022, pp. 1–4.

[23] *IEEE Standard 802.1Q-2018, "IEEE Standard for Local and metropolitan area networks: Bridges and Bridged Networks"*, IEEE Standards Association, July 2018.

[24] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source ieee802. 11 sdr implementation on soc," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–2.

[25] X. Jiao, W. Liu, and M. Mehari. (2019) open-source ieee802.11/wi-fi baseband chip/fpga design. [Online]. Available: https://github.com/open-sdr/openwifi

[26] J. Haxhibeqiri, P. H. Isolani, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, "In-band network monitoring technique to support sdn-based wireless networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 627–641, 2020.

[27] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead, fine-grained end-to-end monitoring of wireless networks using in-band telemetry," in *2019 15th international conference on network and service management (CNSM)*. IEEE, 2019, pp. 1–5.

(https://www.ieee.org/) (http://www.ieee-ies.org/) (https://portale.unipv.it/) (http://webing.unipv.eu/) (https://www.init-owl.de/en/)

April 26 - 28, 2023 - Pavia, Italy

19th IEEE International Conference on Factory Communication Systems (WFCS 2023)

REGISTER NOW

WFCS 2023 is in Pavia, Italy!

After virtually hosting the 2022 edition, Pavia again proposes itself as the venue for the conference, which is supported by IEEE (https://www.ieee.org/) and IES (http://www.ieee-ies.org/) and led by the Technical Committee on Factory Automation (https://tcfa.ieee-ies.org/). The event is organized by the University of Pavia (https://portale.unipv.it/) and by the Institute Industrial IT (inIT) of the Technische Hochschule OWL (https://www.init-owl.de/en/) in Lemgo, Germany.

WFCS is the largest IEEE conference especially dedicated to communications for (industrial) automation systems. Its main aim is to provide a forum for researchers, developers and practitioners to review and discuss most recent trends in the area and share innovative research directions.

Mark the dates in your calendar, and prepare your submissions!